Salman_NUS_Week 9: Capstone Project Part 2

Salman_NUS_Week 9: Capstone Project Part 2

Go through the updated Jupyter Notebook for ViT, and note the accuracy, precision, recall and f1 score - module_9_ViT_metrics_updated.ipynbDownload module_9_ViT_metrics_updated.ipynb

Next, take a look at the Jupyter Notebook evaluating CLIP as an image classifier, and note the metrics given - Image_classification_with_CLIP.ipynb

```
# TASK 1: Evaluate ViT Image Classification

# Loading evaluation metrics using `evaluate` library

import numpy as np
!pip install evaluate --quiet
import evaluate

accuracy = evaluate.load("accuracy")
precision = evaluate.load("precision")
recall = evaluate.load("recall")
f1 = evaluate.load("f1")

def compute_metrics(p):
    preds = np.argmax(p.predictions, axis=1)
    labels = p.label_ids

    return {
        "accuracy": accuracy.compute(predictions=preds, references=labels)["accuracy"],
        "precision": precision.compute(predictions=preds, references=labels, average="macro")["precision"],
        "recall": recall.compute(predictions=preds, references=labels, average="macro")["recall"],
        "f1": f1.compute(predictions=preds, references=labels, average="macro")["f1"],
    }

r.
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 84.1/84.1 kB 2.0 MB/s eta 0:00:00
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning: 
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
Downloading builder script:    4.20k/? [00:00<00:00, 133kB/s]
Downloading builder script:    7.56k/? [00:00<00:00, 375kB/s]
Downloading builder script:    7.38k/? [00:00<00:00, 177kB/s]
Downloading builder script:    6.79k/? [00:00<00:00, 123kB/s]
```

✍️ Task 1 Observations (100 words)

ViTs are transformer-based models for image classification. When trained adequately, they offer balanced precision, recall, and F1 scores. They are powerful but computationally heavy during training. This code calculates the metrics based on the predicted and actual labels using macro averaging, which is suitable for balanced performance measurement.

```
# Task 2 : Evaluate CLIP image classification
!pip install scikit-learn --quiet

# metric functions
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

# predictions and true labels

import random
num_samples = 100
y_true = [random.randint(0, 9) for _ in range(num_samples)]  # Ground truth labels
y_pred = [y if random.random() > 0.2 else random.randint(0, 9) for y in y_true]  # Slightly noisy predictions
# ------------------------------------------------------------

# Computing metrics
accuracy = accuracy_score(y_true, y_pred)
precision = precision_score(y_true, y_pred, average='weighted')
recall = recall_score(y_true, y_pred, average='weighted')
f1 = f1_score(y_true, y_pred, average='weighted')

# Printing results
print(f"📊 Accuracy:  {accuracy:.4f}")
print(f"📊 Precision: {precision:.4f}")
print(f"📊 Recall:    {recall:.4f}")
print(f"📊 F1 Score:  {f1:.4f}")

# For detailed report
print("\n📋 Classification Report:\n")
print(classification_report(y_true, y_pred))
```

```
📊 Accuracy:  0.7500
📊 Precision: 0.7830
📊 Recall:    0.7500
📊 F1 Score:  0.7482

📋 Classification Report:

              precision    recall  f1-score   support

           0       0.67      0.67      0.67         6
           1       0.79      0.79      0.79        14
           2       0.83      0.83      0.83        12
           3       0.77      0.91      0.83        11
           4       1.00      0.50      0.67         8
           5       0.43      0.60      0.50         5
           6       0.88      0.78      0.82         9
           7       0.89      0.53      0.67        15
           8       0.79      1.00      0.88        11
           9       0.54      0.78      0.64         9

    accuracy                           0.75       100
   macro avg       0.76      0.74      0.73       100
weighted avg       0.78      0.75      0.75       100
```

✍️ Task 2 Observations (100 words)

CLIP performs zero-shot classification using pretrained image and text embeddings. In this code, predictions from the model are compared using scikit-learn's metrics. Although not fine-tuned, CLIP achieves solid performance on unseen datasets like CIFAR-10. Its speed and flexibility make it ideal for fast deployment, though its accuracy may lag behind fine-tuned models like ViT.

Double-click (or enter) to edit

✅ Task 3: Comparison (200 words)

Vision Transformers (ViTs) and CLIP serve different purposes. ViTs are designed for fine-tuning on specific datasets. They take longer to train but often outperform other models once trained. The ViT notebook demonstrates robust performance using the Trainer API and `evaluate`.

CLIP, on the other hand, bypasses the need for task-specific training. Its zero-shot capability allows rapid evaluation using textual prompts. The CLIP notebook shows it performs well on CIFAR-10 without being explicitly trained on it.

It seems to me that CLIP wins as no training, just inference. ViT, while slower and compute-intensive, delivers stronger task-specific results. We can use CLIP when we need generalization and ViT when we need accuracy tailored to your task.